

# Mini-project: MultiKnapsack with MinMaxType constraints

---

Nhóm 17

Vi Thành Đạt

Đỗ Trường Mạnh

May 17, 2019

Hanoi University of Science and Technology

1. Mô tả bài toán
2. Mô hình bài toán
3. Dữ liệu đầu vào
4. Chiến lược tìm kiếm
5. Thực nghiệm

## Mô tả bài toán

---

- Có  $N$  items cần xếp vào  $M$  bins
- Item  $i$  ( $i = 1, \dots, N$ )
  - $w_i$ : trọng số 1
  - $p_i$ : trọng số 2
  - $t_i$ : thể loại
  - $r_i$ : lớp
  - $D_i$ : tập bins mà item  $i$  có thể được xếp vào

- Bin  $b$ 
  - $W_b$ : sức chứa 1 (tải tối đa cho trọng số 1)
  - $LW_b$ : tải tối thiểu cho trọng số 1
  - $P_b$ : tải tối đa cho trọng số 2
  - $T_b$ : Số lượng thể loại tối đa cho các items trong bin
  - $R_b$ : số lượng lớp tối đa cho các items trong bin

- Ràng buộc cho mỗi bin  $b$ 
  - C1: Tổng trọng số 1 của các items được xếp vào  $b$  phải lớn hơn hoặc bằng  $LW_b$  và nhỏ hơn hoặc bằng  $W_b$
  - C2: Tổng trọng số 2 của các items được xếp vào  $b$  phải nhỏ hơn hoặc bằng  $P_b$
  - C3: Tổng số thể loại của các items được xếp vào  $b$  phải nhỏ hơn hoặc bằng  $T_b$
  - C4: Tổng số lớp của các items được xếp vào  $b$  phải nhỏ hơn hoặc bằng  $R_b$
- Mục tiêu
  - Tối đa số lượng item xếp được

## Mô hình bài toán

---

- Biến
  - $X_i$ : bin mà item  $i$  được xếp vào,  $D(X_i) = \{0, 1, \dots, N\}$ ,  $X_i = 0 \Rightarrow$  item không được xếp vào bin nào
- Ràng buộc:  $\forall b = 1, \dots, M$ 
  - $LW_b \leq \sum_{i=1}^N (X_i == b) * w_i \leq W_b$
  - $\sum_{i=1}^N (X_i == b) * p_i \leq P_b$
  - $T$
  - $R$



Dữ liệu đầu vào

---

- Nhận xét: Các item có cùng loại và cùng class thì có cùng tập  $D$   
=> Gộp các item này lại thành một nhóm (có  $w$  = tổng các  $w$  các item, có  $p$  = tổng  $p$  các item,  $t$  và  $r$  và  $D$  giống như các item)  
=> 2 kiểu dữ liệu: dữ liệu ban đầu và dữ liệu ghép nhóm

## Chiến lược tìm kiếm

---

- Cách tính violation cho mỗi bin:

$$violation_b = \max(0, _W_b - W_b) + \max(0, LW_b - _W_b) + \max(0, _P_b - P_b) + \max(0, _T_b - T_b) + \max(0, _R_b - R_b)$$

- $_W_b$ : Tổng trọng số 1 của bin b
- $_P_b$ : Tổng trọng số 2 của bin b
- $_T_b$ : Tổng số type trong bin b
- $_R_b$ : Tổng số class trong bin b

# Chiến lược tìm kiếm 01

- Khởi tạo: các item không nằm trong bin nào
- Tìm kiếm: Ưu tiên việc thoả mãn ràng buộc LW của một bin nào đó
  - B1: Duyệt qua tập bin khả dụng xây dựng tập binCandidate có violation nhỏ nhất => Chọn ngẫu nhiên để xét tiếp
  - B2: Xây dựng tập itemCandidate có violationDelta nhỏ nhất => Chọn ngẫu nhiên để tiến hành chuyển sang bin đã chọn
  - Nếu tập itemCandidate của bin được chọn rỗng => loại bin khỏi tập khả dụng

- Chiến lược di chuyển một item từ bin  $bx$  sang bin  $by$  có  $\text{violationDelta} = 0$ : chỉ di chuyển nếu
$$\max(W_{bx} - \_W_{bx}, W_{by} - \_W_{by}) < \max(W_{bx} - \_W'_{bx}, W_{by} - \_W'_{by})$$
 $\Rightarrow$  Ý nghĩa: tạo ra khoảng trống  $W$  lớn hơn cho cơ hội đút thêm item được cao hơn
- Chiến lược reset nếu binCandidate rỗng: bỏ các item khỏi bin có  $\text{violation} > 0 \Rightarrow$  update tập bin khả dụng  
 $\Rightarrow$  tiếp tục tìm kiếm

- Tìm binCandidate gồm các bin có violation thấp nhất, chọn ngẫu nhiên 2 bin.
- Tìm 2 item có thể xếp vào 2 bin nói trên.  
=> Chọn 2 cặp item-bin để sau khi thêm giảm violation nhiều nhất. Nếu item đã ở trong bin, tiến hành đổi bin chứa item ngược lại tiến hành thêm item vào bin.
- Chiến lược cải thiện: Sau khi có danh sách các item mà làm cho violation của các bin = 0, tiến hành tạo ra khoảng trống W còn lại ở mỗi bin lớn nhất mà không làm tăng violation.  
=> Thử xếp các item còn lại vào các bin violation = 0

Thực nghiệm

---



- Chạy thuật toán tìm kiếm trên 2 kiểu dữ liệu:
  - Dữ liệu ban đầu => tìm kiếm lời giải
  - Dữ liệu ghép nhóm => tìm kiếm lời giải => phân rã thành item tách biệt => cải thiện bằng tìm kiếm tiếp trên lời giải đã phân rã

## Kết quả thực nghiệm

	Bộ 1000 item	Bộ 3000 item
Search1	302	1961
Search1 + dữ liệu nhóm	547	2555
Search2	Chạy rất lâu	Chạy rất lâu
Search2 + dữ liệu nhóm	338	645

Cảm ơn đã lắng nghe